

6장. 데이터 타입

👤 생성자	👤 한빈 이
🕒 생성 일시	@2024년 2월 17일 오전 12:29
🏷️ 태그	공부

6장. 데이터 타입

- 데이터 타입(data type) : 값의 종류
- 자바스크립트(ES6)는 **7개**의 데이터 타입을 제공
- 원시 타입(primitive type), 객체 타입(object/reference type)으로 분류

구분	데이터 타입	설명
원시 타입	숫자 ^{number} 타입	숫자. 정수와 실수 구분 없이 하나의 숫자 타입만 존재
	문자열 ^{string} 타입	문자열
	불리언 ^{boolean} 타입	논리적 참(true)과 거짓(false)
	undefined 타입	var 키워드로 선언된 변수에 암묵적으로 할당되는 값
	null 타입	값이 없다는 것을 의도적으로 명시할 때 사용하는 값
	심벌 ^{symbol} 타입	ES6에서 추가된 7번째 타입
객체 타입		객체, 함수, 배열 등

6.1 숫자 타입

1. 숫자 타입

```
var integer = 10;
```

```
var double = 10.12;
```

```
var negative = -20;    ⇒ 모두 숫자 타입
```

- 자바스크립트의 숫자 타입은 정수만을 위한 타입이 없고 모든 수를 **실수**로 처리

6.2 문자열 타입

1. 따옴표로 감싸지 않은 hello를 식별자로 인식

⇒ `var string hello; // ReferenceError : hello is not defined`

6.3 템플릿 리터럴

1. 템플릿 리터럴은 백틱(` `)을 사용하여 표현

6.3.1 멀티라인 문자열

이스케이프 시퀀스	의미
<code>\0</code>	Null
<code>\b</code>	백스페이스
<code>\f</code>	폼 피드 ^{Form Feed} : 프린터로 출력할 경우 다음 페이지의 시작 지점으로 이동한다.
<code>\n</code>	개행 ^{LF, Line Feed} : 다음 행으로 이동
<code>\r</code>	개행 ^{CR, Carriage Return} : 커서를 처음으로 이동
<code>\t</code>	탭(수평)
<code>\v</code>	탭(수직)
<code>\uXXXX</code>	유니코드, 예를 들어 <code>'\u0041'</code> 은 'A', <code>'\uD55C'</code> 는 '한', <code>'\u{1F600}'</code> 는 🤪이다.
<code>\'</code>	작은따옴표
<code>\"</code>	큰따옴표
<code>\\</code>	백슬래시

[예제 06-11]

```
var template = `


  <li><a href="#">Home</a></li>
</ul>`;
```

```
console.log(template);
```

출력결과

- 이스케이프 시퀀스를 사용하지 않고도 줄바꿈이 허용, 모든 공백도 있는 그대로 적용

```
<ul>
  <li><a href="#">Home</a></li>
</ul>
```

6.3.2 표현식 삽입

1. [예제 06-13]

```
var first = 'Ung-mo';
var last = 'Lee';
console.log(My name is ${first} ${last}.); // My name is Ung-mo Lee.
```

- 표현식을 삽입하려면 \${ } 으로 표현식을 감싼다.
- 표현식 평가 결과가 문자열이 아니더라도, 문자열로 타입이 강제로 변환되어 삽입

6.4 불리언 타입

1. Boolean 타입 : 논리적 참(true), 거짓(false)

2. 참과 거짓으로 구분되는 조건에 의해 프로그램 흐름을 제어하는 조건문에서 자주 사용

6.5 undefined 타입

1. 자바스크립트 엔진이 변수를 초기화하는 데 사용하는 undefined, 개발자가 의도적으로 변수에 할당한다면 undefined의 본래 취지와 어긋남 + 혼란을 줄 수 있으므로 권장 X
 - 변수에 값이 없다는 것을 명시하고 싶을 때 → undefined가 아닌 null을 할당

6.6 null 타입

1. 프로그래밍 언어에서 null : 변수에 값이 없다는 것을 의도적으로 명시(의도적 부재)할 때 사용
2. 변수에 null 할당하는 것 : 변수가 이전에 참조하던 값을 더 이상 참조하지 않겠다는 의미
 - ➡ 이전에 할당되어 있던 값에 대한 참조를 명시적으로 제거

6.7 심벌 타입

1. 심벌(symbol)은 ES6에서 추가된 7번째 타입, 변경 불가능한 원시 타입의 값
 - 다른 값과 중복되지 않는 유일무이한 값
 - ➡ 주로 이름이 충돌할 위험이 없는 객체의 유일한 프로퍼티 키를 만들기 위해 사용
2. 심벌은 Symbol 함수를 호출해 생성

6.8 객체 타입

1. 자바스크립트는 객체 기반의 언어, **자바스크립트를 이루고 있는 모든 것이 객체**라는 것.

6.9 데이터 타입의 필요성

6.9.1 데이터 타입에 의한 메모리 공간의 확보와 참조

1. 자바스크립트 엔진은 데이터 타입, 즉 **값의 종류에 따라 정해진 크기의 메모리 공간을 확보**함.
즉, 변수에 할당되는 값의 데이터 타입에 따라 확보해야 할 **메모리 공간의 크기가 결정**됨

6.9.2 데이터 타입에 의한 값의 해석

[데이터 타입이 필요한 이유]

이유1. 값을 저장할 때 확보해야 하는 **메모리 공간의 크기**를 결정하기 위해

이유2. 값을 참조할 때 한 번에 읽어 들여야 할 **메모리 공간의 크기**를 결정하기 위해

이유3. 메모리에서 읽어 들인 **2진수를 어떻게 해석**할지 결정하기 위해

6.10 동적 타이핑

6.10.1 동적 타입 언어, 정적 타입 언어

1. C, Java 같은 **정적 타입**(static/strong type) **언어**는 변수 선언할 때 변수에 할당할 수 있는 값의 종류, 즉 데이터 타입을 사전에 선언해야함.
 - 이를 명시적 타입 선언(explicit type declaration)이라 함.
 2. 정적 타입 언어는 변수의 타입을 변경할 수 없으며, 변수에 선언한 타입에 맞는 값만 할당 가능.
-
3. 자바스크립트는 정적 타입 언어와 다르게, 변수 선언할 때 타입을 선언하지 않음.
다만 **var**, **let**, **const** 키워드를 사용해 변수를 선언할 뿐.

➡ 자바스크립트의 변수는 선언이 아닌 할당에 의해 타입이 결정(타입 추론) 된다. 그리고 재할당에 의해 변수의 타입은 언제든지 동적으로 변할 수 있다. 이러한 특징을 동적 타이핑이라 하며, 자바스크립트는 정적 타입 언어와 구별하기 위해 **동적 타입 (dynamic/weak type)언어**라 한다.

6.10.2 동적 타입 언어와 변수

1. 장점

동적 타입 언어는 변수에 어떤 데이터 타입의 값이라도 자유롭게 할당할 수 있다.

2. 단점

2-1. 복잡한 프로그램에서는 변화하는 변수 값을 추적하기 어려울 수 있다.

2-2. 변수의 타입이 고정되어 있지 않고 동적으로 변하는 동적 타입 언어의 변수는 값의 변경에 의

해 타입도 언제든지 변경될 수 있다

- 동적 타입 언어는 유연성(flexibility)은 높지만 신뢰성(reliability)은 떨어짐.

따라서 변수를 사용할 때 **주의할 사항**은 다음과 같다.

- 변수는 꼭 필요한 경우에 한해 제한적으로 사용한다.
- 변수의 유효 범위(스코프)는 최대한 좁게 만들어 변수의 부작용을 억제해야 한다.
- 전역 변수는 최대한 사용하지 않도록 한다.
- 변수보다는 상수를 사용해 값의 변경을 억제한다.
- 변수의 이름은 변수의 목적이나 의미를 파악할 수 있도록 네이밍한다.

➡ 가독성이 좋은 코드가 좋은 코드다.